# Machine vs Humans in a Cursive Script Reading Experiment without Linguistic Knowledge*

Marc PARIZEAU† and Réjean PLAMONDON‡

printed July 8, 1997

## Abstract

*This paper presents an overview of a dynamic cursive script recognition approach that uses no linguistic constraints. This approach seeks to recognize in cursive script, morphologically and pragmatically coherent sequences of character hypotheses. Its performance is compared with the performance of the best available cursive script recognizers* — humans — *in a reading experiment where linguistic knowledge is useless.*

*The recognition method uses fuzzy-shape grammars to model the morphological characteristics of conventional letters. These models, called* allographs, *can be viewed as basic (a priori) knowledge for developing a multi-writer recognition system. Character hypotheses are segmented within a cursive word using a parser for these grammars. Character sequences are then constructed from these segmentation hypotheses using local adjacency constraints also modeled by fuzzy-shape grammars.*

*Two experiments are conducted on a test database containing a handwritten cursive text 600 characters in length written by ten different writers. First, a reading experiment with ten human readers yields an average character recognition rate of* 96.0%. *Second, a test of the recognition system gives an average character recognition rate of between* 84.4% *to* 91.6%, *depending on whether only the first (best) character sequence output of the system is considered or if the best of the top ten is accepted. This result is achieved without any writer dependent tuning. Moreover, results show that system performances are highly correlated with human performance.*

## I Introduction

Cursive script recognition systems can rely essentially on three types of knowledge: morphological, pragmatic and linguistic. *Morphological* knowledge refers to everything that is known about the shapes of cursive letters.

Figure 1: *Trace samples of lower case character a.*

*Pragmatic*[1] knowledge refers to what is known about how to spatially arrange cursive letters into words, phrases and paragraphs. *Linguistic* knowledge concerns the language that is used to convey the message represented in handwriting (i.e English, French, etc.).

The recognition system which is compared in this paper with human readers, is based exclusively on morphological and pragmatic knowledge. It was developed in an effort to circumvent the usual limited vocabulary constraint that is common with most reported cursive script recognition system. The idea is to build writer-independent models for the principal shapes of conventional cursive letters — each of these corresponding to a distinct *allograph* — and to design an algorithm that can recognize portions of a cursive word that match these models.

This approach lies on the assumption that there exists a limited number of ways (models) to write each letter of the alphabet, even in the context of cursive script. This can be supported by the fact that cursive script is taught to schoolchildren and that, although different teaching methods can be found in different schools or regions [1], literate humans are generally capable (to some extent) of reading cursive script that they have never encountered before. This last statement, of course, makes the supplementary assumption that the human who wrote the script wanted to be recognized!

Now based on this assumption, a cursive trace of a particular allograph is an approximation of an ideal model. Thus, to recognize that a particular trace is an instance of a particular allograph, we need to model what is characteristic of that allograph, not what is characteristic of the writer who produced the trace. For example, consider the trace samples of Figure 1 for the lower case character *a*. All these samples can be viewed as traces of a single allograph model consisting of a *c-shape* horizontally concatenated with a dotless *i-shape* somewhat like the first (leftmost) trace of the figure. The other traces of the fig-

*This work was published in *Proc. of the 12th International Conference on Pattern Recognition*, Jerusalem, October 9-13, vol. II, pp. 93-98, 1994. It was supported in part by NSERC Canada under grant OGP000915, and in part by FCAR Québec under grant CRP2667. Marc Parizeau received scholarships from NSERC and FCAR.

†Marc Parizeau was with the Département de génie électrique et de génie informatique, École Polytechnique de Montréal. He is now with the Département de génie électrique et de génie informatique, Université Laval, St-Foy (PQ), Canada, G1K 7P4. E-mail: *parizeau@gel.ulaval.ca*

‡Réjean Plamondon is with the Département de génie électrique et de génie informatique, École Polytechnique de Montréal, C.P. 6079 – Succ. "A", Montréal, Canada, H3C 3A7.

[1]This definition of pragmatic knowledge should not be confused with the definition used by linguists, who refer to linguistic knowledge induced by common sense.

ure can be "explained" by this principal allograph of letter *a*, either by adding a ligature (in front or at the end), by degenerating the *i-shape* into an *e-shape* or by tilting the *c-shape* or *i-shape* to the right.

To define such robust allograph models, powerful tools are required. They are discussed briefly in the next section together with a complete system overview. Section III then presents the main topic of this paper, that is, the performance comparison of that system with human readers in a linguistic free context. The object of that comparison being to evaluate to what extent this system is capable of recognizing cursive script without using any linguistic knowledge.

# II System Overview

The developed system is composed of three principal processes as illustrated in Figure 2. The first process is a **primitive extraction** step that segments the *handwriting components*[2] of a word into a sequence of attributed *cursive primitives*. This segmentation process is based on a *handwriting model* that represents a component by a sequence of characteristic points linked by segments of constant curvature (circular arcs). For example, Figure 3 illustrates a cursive trace of two components for word 'axe' (French for 'axis'). The small squares in that figure represent the digitized points of the components, the + and × marks correspond to the characteristic points of the model, and the continuous curve between two successive characteristic points is a circular arc obtained by a fitting procedure. The model defines the characteristic points as the morphologically pertinent points of the components for recognition purpose. Here the characteristic points correspond to the horizontal and vertical local extrema (+ marks) of the trace and to the inflexion points (× marks).

From this representation of the handwriting components, a *cursive primitive* is extracted for every characteristic point (for the example of figure 3, there are 22 characteristic points and, thus, 22 primitives). A primitive spans the portion of the trace that goes from the previous characteristic point to the next one or, in other words, it is associated with two successive circular arcs. Each primitive is attributed, that is, it is defined by a set of attributes of two types: attachment points and properties. The attachment points of the primitive are the three characteristic points covered by the primitive (previous, middle and next). There are seven properties [3]: a measure of discontinuity at the middle attachment point, a measure of tilt at each of the three attachment points, a measure of curveness[3] for each of the two circular arcs associated with the primitive and a unique index number that identifies the primitive within the sequence of primitives.

[2]The term *component* [2] is used to designate the portion of the written trace between a pendown and a penlift (while the pen is in contact with the paper).

[3]The term *curveness* is used to designate a heuristic measure of curvature.

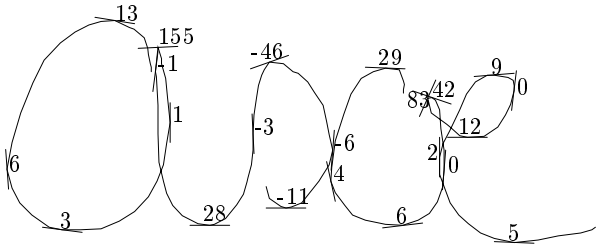


Figure 4: *Illustration of tilt and discontinuity properties.*

The second process of the system is an **allograph segmentation** step that analyses the set of all cursive primitives extracted by the previous process and determines which subsets of these primitives respect the conditions of the *allograph models*. This second segmentation step generates recognition hypotheses for morphologically coherent allograph traces. The models are hand-generated using a specially designed programming language called $\mathcal{HAD}$ ($\mathcal{H}$ierarchical $\mathcal{A}$llograph $\mathcal{D}$escription) [4] that enables the specification of fuzzy-shape grammars [5], that is, attributed grammars with production rules governed by fuzzy logic.

The modeling strategy for building allographs starts with a basic handwriting element that possesses all the attributes of the cursive primitive. This element serves as a terminal symbol for all grammars. It represents a perfectly general piece of a component that spans three characteristic points. The first phase of the strategy is to create several classes of this element by defining more specific elementary shapes. This is simply done with a production rule that applies a fuzzy-membership function on certain properties of the basic handwriting element. For example, Figure 4 illustrates the tilt and discontinuity properties for each characteristic point of a trace: *tilt* is represented by the angle of a line segment and *discontinuity* is represented by a number in the range $[-180, 180]$ where 0 means very continuous and $-180$ or 180 means very discontinuous. Thus, for example, a given cursive primitive could belong to the class of "continuous" and "horizontal" elements, that is, would represent essentially a continuous horizontal displacement. It must be emphasized here that such a class is created by a membership function (fuzzy-threshold) and thus defines a fuzzy-set. This implies that a given primitive may belong to several classes of elements, although not necessarily with the same grade of membership.

The second phase of the modeling strategy is to assemble with other production rules, basic shapes like *c-shapes*, *i-shapes*, *loops*, *dots*, *t-crossings*, etc..., using combinations of pertinent classes of handwriting elements and fuzzy-thresholds on their attributes. These basic shapes are nothing less than sub-allograph models and, again, define fuzzy-sets.

The third and final phase of the modeling strategy is to assemble the allographs themselves using combinations of pertinent basic shapes, pertinent classes of handwriting

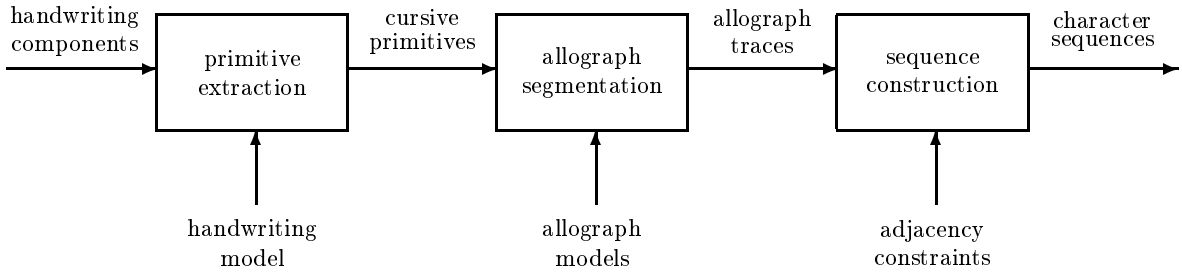


Figure 2: *Block diagram of system.*

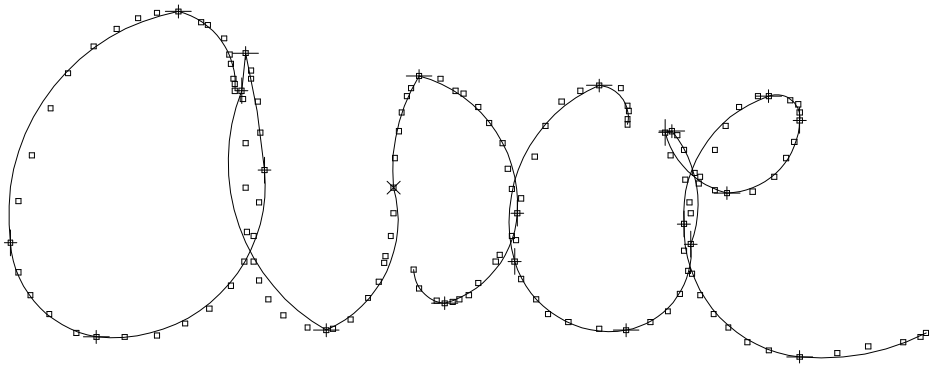

Figure 3: *Illustration of the primitive extraction process for the trace of word /axe/.*

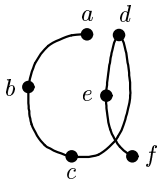


Figure 5: *Illustration of an allograph of lower case letter a.*

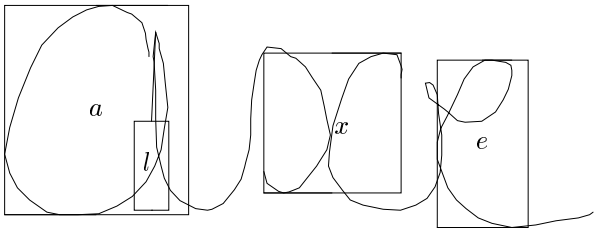


Figure 6: *Examples of segmented traces in word* axe.

elements and several *morphological characteristics* (MC) of the allographs. For instance, Figure 5 illustrates the principal allograph model of lower case letter *a* with its different attachment points. It consists essentially in two sub-allographs: a *c-shape* combined with an *i-shape* or an *e-shape*. Using its attachment points, it is possible to define relative measures of height or width that characterize morphologically this allograph. Indeed, simply by moving point *f* (and point *e* with it) up to approximately point *d*, we could get an allograph of the letter *o*. Similarly, if point *f* is moved down sufficiently, we can get an allograph of the letter *q*. Thus, a first MC is the height of the final ligature relative to the vertical position of the *i-shape*. Likewise, if point *d* is moved up sufficiently relative to point *a*, then we get an allograph of the letter *d*. A second MC thus measures the height of the *i-shape* relative to the height of the *c-shape*. Finally, if the gap between the *c-shape* and the *i-shape* widens, then the *a* allograph would become invalid. Hence, a third MC can take this gap into account.

Using this strategy, some 18 classes of elements, 22 sub-allograph symbols and 54 allograph models, for lower-case letters of the Roman alphabet, were created [4] ($\approx$ 2 000 lines of $\mathcal{HAD}$). The corresponding grammars are used by a parser to recognize morphologically coherent allograph traces within the set of cursives primitives [5].

Finally, the third process of the system is a **sequence construction** that select pragmatically consistent combinations of segmented *allograph traces* [6]. All allograph models are designed in such a way that each trace is segmented with four attachment points corresponding to the lower left and upper right corners of the bounding box that encloses the main body of the trace, and to the highest and lowest points of the trace (for ascenders and descenders). For example, Figure 6 shows the bounding boxes of four segmented traces including the traces of letters *a*, *x* and *e*. The forth trace that represents an *l*, is an example of a morphologically coherent recognition hypothesis which is not, however, pragmatically consistent with the other segmentation hypotheses. This process thus analyses the spatial adjacency for pairs of allograph traces, using both a

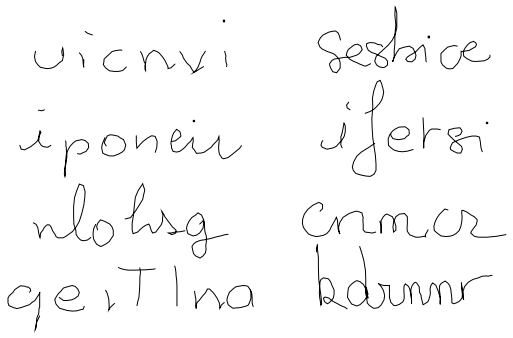

Figure 7: *Samples of cursive random letter sequences.*

horizontal constraint that measures the coherence of their horizontal spacing, and a vertical constraint that measures the quality of their vertical alignment. With these adjacency constraints, an allograph segmentation graph is built and the best paths in that graph correspond to the best *character sequences* recognized by the system.

# III Experimental Results

To evaluate and compare the performance of the system overviewed in the previous section, 100 character sequences (of length between 5 and 7 characters) were generated randomly while respecting the letter occurrence frequencies of a 32,000-word French dictionary. Then, 10 different writers were selected from a pool of graduate students to write these sequences using their natural, although clean, handwriting style. No constraints were imposed on the writers except to maintain what they considered good legibility. Figure 7 shows a few samples taken from this test dataset. As can be seen from these samples, some of the writers (in fact 2) naturally detach each letter from the next.

Data acquisition was conducted using a *PenPad 300* digitizing tablet from Pencept Inc. with a resolution of 0.001″, a sampling frequency of 100 Hz and a precision (as specified by the manufacturer) of 0.005″.

Because the goal of 100% recognition is probably not realistic, an experiment was designed to establish a more suitable reference performance for the constructed dataset. This reference performance is that of the best known cursive script recognizers: humans. A second group of 10 volunteers (different from the writers) was thus chosen from a pool of graduate students and research staff, some of them experts in pattern recognition and image processing. All character sequences in the dataset ($10 \times 100$ character sequences) were assigned randomly to the volunteers (100 character sequences each) with the constraint that each reader would be presented with 10 samples from each writer. The samples of the dataset were printed on sheets of paper (25 samples/sheet) and the volunteers were asked simply to transcribe into an ASCII file what they could read for each sample. The only hint that they were given was that the samples contained only lower case-letters of the Roman alphabet.

Table 1 summarizes the error rates of the human readers (— indicates no error). These error rates are computed using the Wagner-Fischer string-to-string editing distance [7] with unit cost for insertion, deletion and substitution. They should be interpreted as distance measures between the correct character sequence and the response given by the human reader.

Looking at the rightmost column of that table, we can immediately observe that average human error rates on that dataset have varied from a minimum of 2% to a maximum of 6.9% for a global average error of 4%. In other words, we should probably not seek for our system on this dataset, a performance higher than 96% or even 93%. Moreover, if we consider the worse reader/writer combination (for reader #2 and writer #6), we can observe an error rate as high as 21.7%.

Looking at the bottom line of that table, we also observe that the handwriting of some writers seem to be much more difficult to read than others. Indeed, average error rates for writers range from 0.8% to 8.9%, that is, a tenfold difference between the most legible and the worst. It is also interesting to note that writer #1 and #7, who are the two writers that naturally detach each letter from the next, are not "the" most legible writers, although they are more legible than the average one.

For the second experiment, the same dataset was used to evaluate the performance of the recognition system. Recognition rates for each writer are given in Table 2 (recognition rates correspond here to the complement of the error rates defined in the first experiment). The first line of that table gives the results when considering only the character sequence output ranked first by the system. The next four lines of this table list the results when the best of the first 2, 3, 5 and 10 sequence outputs are considered respectively.

The average recognition rate for the system thus varies from 84.4% to 91.6%, depending on whether only the sequence ranked first is considered, or if the best of the top 10 sequences is accepted. It should be noted here that the recognition system is multi-writer, in the sense that system parameters are the same for all writers, and that writers #4, #6, #8 and #10 were completely unknown to the system since they didn't participate in the construction of any training datasets (the system was mainly trained using an independent dataset of isolated cursive letters).

Looking at the system's performance for these "unknown" writers, we can observe that it is somewhat inferior (except for writer #10) to what was achieved for the other writers. However, it should also be noted that three of these four writers collected the worst human performances. The two bottom lines of Table 2 compare the performance of the system with the performance of the humans. They first show that human and system performances are highly correlated (by a factor of 0.92) and, second, that if average human performance (next to last line) is considered the new goal and that the best of the top ten sequence outputs is acceptable, then the residual error of the system,

Table 1: *Error rates of human readers (in %).*

| Reader | Writer 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | Average |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1.6 | — | 5.5 | 11.5 | 6.0 | 4.8 | 3.2 | 9.8 | — | — | 4.2 |
| 2 | 4.9 | 1.6 | 16.4 | 4.8 | 3.2 | 21.7 | 4.8 | 1.7 | 6.8 | 3.1 | 6.9 |
| 3 | 3.2 | — | 1.6 | 5.3 | — | 9.8 | 1.5 | 9.4 | 5.3 | 1.6 | 3.8 |
| 4 | — | 3.4 | 6.2 | 6.5 | 1.7 | 15.3 | — | 4.6 | 1.7 | 8.3 | 4.7 |
| 5 | — | — | 5.0 | — | 3.1 | 1.5 | 3.6 | 8.2 | 1.6 | — | 2.3 |
| 6 | 1.6 | — | 9.1 | 3.2 | 4.9 | 10.0 | 3.1 | 12.1 | — | 3.2 | 4.7 |
| 7 | — | 1.7 | 3.3 | 3.4 | 3.4 | 10.6 | 3.1 | 6.6 | 3.2 | 1.7 | 3.8 |
| 8 | 6.3 | 1.6 | 1.6 | 3.2 | 5.2 | 6.6 | 1.7 | 4.9 | 3.2 | 8.1 | 4.2 |
| 9 | — | — | 1.8 | 4.8 | — | 4.9 | 1.6 | 3.5 | 1.7 | 1.6 | 2.0 |
| 10 | 1.6 | — | 3.2 | 6.7 | 1.6 | 5.0 | 3.2 | 12.5 | 1.6 | — | 3.4 |
| Average | 1.9 | 0.8 | 5.4 | 4.9 | 2.9 | 8.9 | 2.6 | 7.3 | 2.5 | 2.8 | 4.0 |

Table 2: *Recognition rates of the system (in %).*

| accepted ranks | writer #1 | #2 | #3 | #4 | #5 | #6 | #7 | #8 | #9 | #10 | average rate |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 91.0 | 91.7 | 84.5 | 78.5 | 92.2 | 68.2 | 82.7 | 75.5 | 89.6 | 89.9 | 84.4 |
| 1-2 | 93.1 | 95.8 | 88.9 | 81.9 | 94.8 | 73.4 | 85.5 | 80.1 | 94.1 | 93.8 | 88.1 |
| 1-3 | 93.1 | 96.7 | 90.5 | 83.4 | 95.8 | 75.2 | 85.8 | 83.0 | 95.3 | 94.6 | 89.3 |
| 1-5 | 93.5 | 97.7 | 93.1 | 85.3 | 95.9 | 77.5 | 86.9 | 85.5 | 96.1 | 95.4 | 90.7 |
| 1-10 | 93.5 | 98.2 | 93.8 | 87.1 | 96.2 | 78.6 | 87.3 | 88.4 | 96.7 | 96.1 | 91.6 |
| humans | 98.1 | 99.2 | 94.6 | 95.1 | 97.1 | 91.1 | 97.4 | 92.7 | 97.5 | 97.2 | 96.0 |
| residues | 4.6 | 1.0 | 0.8 | 8.0 | 0.9 | 12.5 | 10.1 | 4.3 | 0.8 | 1.1 | 4.4 |

given on the last line of Table 2, varies from 0.8% to 12.5% for an average of 4.4%. Furthermore, they also show that for writers #2, 3, 5, 9 and 10, this residual error is less than 1.1%.

The choice of selecting the top ten results is of course arbitrary. However, the important thing is that most of the correct characters be contained in these first few outputs so that a list of morphologically and pragmatically coherent alternatives of limited length can always be produced. Then, any available linguistic knowledge can be used by higher decision processes. For instance, in a typical limited vocabulary application, efficient and well-known search techniques can be used to scan a dictionary to find words that approximately match the list of character sequences produced by the system [8, 9].

# IV Conclusion

The idea behind the recognition approach presented in this paper is to separate two distinct, though complementary, problems: the *recognition* of a set of graphics symbols (i.e. a given alphabet), and the *reading* of a message coded with these symbols (i.e. a text written in a given language). The advantage of such a separation is that solutions to these two fundamental and difficult problems can be optimized independently and, eventually, be integrated into a global, multiple approach, handwriting reading system.

Of course, this is not to say that linguistic knowledge cannot or should not be integrated into recognition processes, but rather that recognition should not be restricted to the linguistic knowledge. In that sense, systems that restrict recognition to a limited and fixed vocabulary are useful only for special applications. The use of letter n-grams is less restrictive and could obviously benefit the sequence construction process. But the object of this work was to demonstrate the feasibility of recognizing cursive script, at least partially, without using any linguistic context.

The allograph models developed for testing the proposed approach have demonstrated this feasibility even though they are far from perfect. Indeed, for half the writers, when considering the best of the top ten sequence outputs of the system, the results obtained were almost as good as those of very proficient human readers, while for the other half they were somewhat inferior. After examining some of the recognition errors, and without changing the modeling strategy, it is clear that some allographs could be recoded to make them more robust and that some new allographs should be added. This is, however, a time-consuming process even though it has to be done only once.

# V Acknowledgements

The authors would like to thank all the volunteers who contributed samples of cursive handwriting, as well as those who participated in the human reading experiment. This work was supported partially by grant OPG-0915 from NSERC Canada and by grant ER-1220 from FCAR Québec.

# References

[1] Suen C.Y., "Handwriting Education — A Bibliography of Contemporary Publications", *Visible Language*, vol. 9, p. 145-158, 1975.

[2] Plamondon R., Maarse F.J., "An Evaluation of Motor Models of Handwriting", *IEEE trans. on Systems, Man and Cybernetics*, vol. SMC-19, no. 5, pp. 1060-1072, 1989.

[3] Parizeau M., Plamondon R., "A Handwriting Model for Syntactic Recognition of Cursive Script", *Proc. of the* $11^{th}$ *International Conference on Pattern Recognition*, The Hague, August 31 to September 3, vol. II, pp. 308-312, 1992.

[4] Parizeau M., "Reconnaissance d'écriture cursive par grammaires floues avec attributs: étape vers la conception d'un bloc-notes électronique", Ph.D. Thesis, Ecole Polytechnique de Montréal, 1992.

[5] Parizeau M., Plamondon R., Lorette G., "Fuzzy-Shape Grammars for Cursive Script Recognition", *Advances in Structural and Syntactic Pattern Recognition*, H. Bunke (editor), World Scientific Publishing, pp. 320-332, 1993.

[6] Parizeau M., Plamondon R., "Allograph Adjacency Constraints for Cursive Script Recognition", *Proc. of the Third International Workshop on Frontiers in Handwriting Recognition*, Buffalo, May 25-27, pp. 252-261, 1993.

[7] Wagner R.A., Fischer M.J., "The String-to-String Correction Problem", *Journal of the ACM*, vol. 21, no. 1, pp. 168-173, 1974.

[8] Bozinovic R.M., Shrihari S.N., "A String Correction Algorithm for Cursive Script Recognition", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. PAMI-4, p. 655-663, novembre 1982.

[9] Wells C.J., Evett L.J., Whitby P.E., Whitrow R.J., "The use of Orthographic Information for Script Recognition", in *Computer Processing of Handwriting*, R. Plamondon & C.G. Leedham (editors), World Scientific Publishing, p. 273-289, 1990.